CLAIMS

1. Method for processing a complex request addressed to at least one SNMP agent (5) of a resource machine (2b) of a computer system (1) from a complex protocol manager (4) of an application machine (2a), the application (2a) and resource (2b) machines communicating through a network (3), each agent (5) managing attribute tables belonging to the resource machine (2b), the instances of the tables being referenced by identifiers comprising indexes, characterized in that it consists of:

- transforming a filter (F1) derived from a complex request from the manager (4) of the application machine (2a) into a simplified filter (F2) comprising only conditions on indexes, the filter (F2) corresponding to the following matching characteristics: the filter (F2) lets through all the SNMP requests whose responses could verify the filter (F1), but filters out all the SNMP requests whose responses cannot in any way verify the filter (F1);
- limiting the SNMP requests to those that comply with the filter (F2);
- transmitting said limited SNMP requests to the SNMP agent (5) of the resource machine (2b) through the network (3);
- applying the filter (F1) to the responses obtained to the SNMP requests;

the method making it possible to process said complex request and to optimize the number of the SNMP requests transmitted through the network (3).

2. Method according to claim 1, characterized in that it consists of:

1) Transforming the filter (F1) derived from the complex request into a simplified filter (F2);
2) determining the first potential instance that verifies the simplified filter (F2); the identifier just below the identifier of the potential instance determined is called the test identifier;
3) finding, using an SNMP request, the instance of the table having as its identifier the one that follows the test identifier. If no instance is found, the processing method is terminated. If an instance is found, the instance found is called the solution instance;
4) applying the complex filter (F1) to the solution instance; if the instance verifies the filter (F1), it is part of the response to the complex request processed;

5) determining the first potential instance whose identifier is higher than the identifier of the solution instance and that verifies the simplified filter (F2). If no instance is found, the processing method is terminated. If an instance is found, the identifier that is just below the identifier of the potential instance is called the test identifier and the method resumes with the third step.

3. Method according to claim 2, characterized in that it consists of obtaining, in the first step, the simplified filter with the form:

(OR
  (AND
    condition on index 1: $C1_{(1)}$
    condition on index 2: $C2_{(1)}$
    ...
    condition on index n: $Cn_{(1)}$
)
  ...
  (AND
    condition on index 1: $C1_{(i)}$
    condition on index 2: $C2_{(i)}$
    ...
    condition on index n: $Cn_{(i)}$
)
  ...
).

4. Method according to either of claims 2 and 3, characterized in that, if in the first step, after simplification, the filter is reduced to:

- only the TRUE condition, the table is scanned in its entirety;
- only the FALSE condition, no instance can work.

5. Method according to any of claims 2 through 4, characterized in that, in order to obtain the simplified filter F2, it consists of immediately verifying whether the complex filter responds to rules defining filters that are not verified by any instance.

6. Method according to any of claims 1 through 5, characterized in that, in order to obtain a simplified filter F2, it consists of:

- transforming the complex filter into a combination of conditions on the attributes joined by the logical operators HAND, OR and NOT;
- pushing the NOT operators to the leaves and deleting the double NOTs (NOT NOT);
- deleting the conditions X affecting the attributes that are not indexes;
- simplifying the resulting operations;
- factoring the nested ANDs and ORs;
- gathering the conditions related to the same index;
- gathering all the ORs at the route of the filter and simplifying again.

7. Method according to claim 6, characterized in that in order to delete the conditions X, it consists of replacing the conditions X and NOT X with the constant TRUE.

8. Method according to either of claims6 and 7, characterized in that in order to simplify the operations, it consists of:

- replacing the AND and OR tests having only one operand with this operand;
- replacing the AND operations containing only TRUE operands with the constant TRUE and the OR operations containing only FALSE operands with the constant FALSE;
- removing the TRUE conditions from the other AND operations and the FALSE conditions from the other OR operations;
- replacing the OR operations containing at least one TRUE operation with the constant TRUE and the AND operations containing at least one FALSE operand with the constant FALSE;
- replacing the conditions that are always TRUE or FALSE with the constant TRUE or FALSE;

all of these simplification operations being applied as many times as it is possible to do so.

9. The method according to any of claims 2 through 8 characterized in that, in the second step, it consists of concatenating the first a value that verifies $C1_{(i)}$ with the first value that verifies $C2_{(i)}$, and so on up to $Cn_{(i)}$, in order to obtain the zero local potential instances $I1\_0_{(i)}.I2\_0_{(i)}. \ldots In\_0_{(i)}$, the first possible value without a condition on a given index being the null value, the potential instance corresponding to the smallest of the zero local potential instances.

10. Method according to claim 9, characterized in that, in the fifth step, it consists of performing, for any i and as long as the index p is greater than 0 or as long as no instance searched for has been found, the following operations:

If there exists a $Jp_{(i)} > Ip$ that verifies the condition $Cp_{(i)}$, then the local potential instance is formed in the following way:

- for any index $k < p$, we take the value Ik with I1.I2. ... .In being the identifier of the solution instance;
- for the index p, we take the value $Jp_{(i)}$;
- for any index $k > p$, we take the value $Ik\_0_{(i)}$;

Otherwise p takes the value p-1 and the method repeats the above operations, the potential instance corresponding to the smallest of the local potential instances obtained.

11. Method according to any of claims 2 through 10, characterized in that in the second and fifth steps, it consists of obtaining the test identifier from the identifier of the potential instance, by subtracting one from its last number if the latter is different from 0, or by deleting this last number if it is null.

12. System for processing a complex request addressed to at least one SNMP agent (5) of a resource machine (2b) of a computer system (1) from a complex protocol manager (4) of an application machine (2a), each agent (5) managing attribute tables belonging to the resource machine (2b), the instances of the tables being referenced by identifiers comprising indexes, the system comprising an integrating agent (6) that makes it possible to implement the processing method according to any of claims 1 through 11.

ADD A6)